

AWS IoTを使った cookpad storeTV端末の管理

IoT@Loft
2020/08/19

About me

- Kenji Abe
- Cookpad Inc.
- cookpadLive, storeTV
- Google Developers Experts for Android, Kotlin
- Twitter: @STAR_ZERO

cookpad storeTVとは

storeTV

- 全国スーパーに設置
 - 農産、畜産、水産
 - 現在約17,000台
- 売り場に連動した料理動画
- スーパーでレシピが決まる



これまでの課題

これまでの課題

- 他社のMDMサービス
 - コスト
 - 管理画面との連携
 - 安定性
 - 修正のスピード

AWS IoTとは

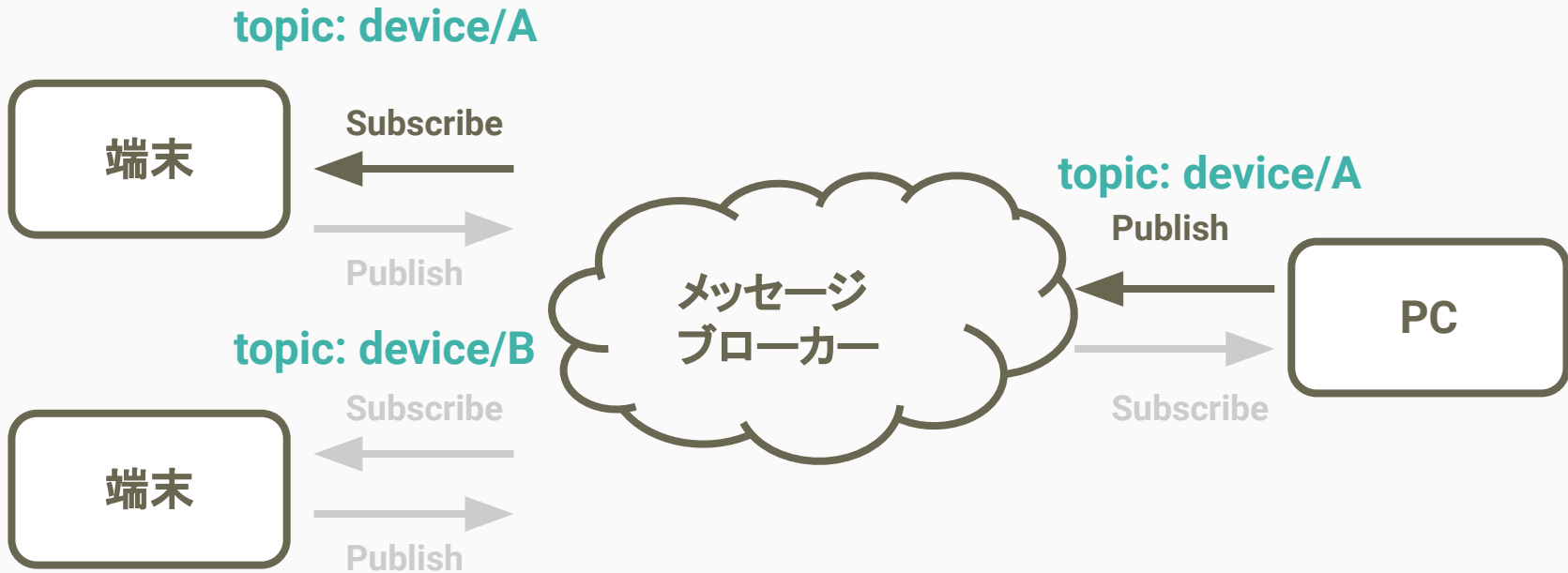
AWS IoTとは

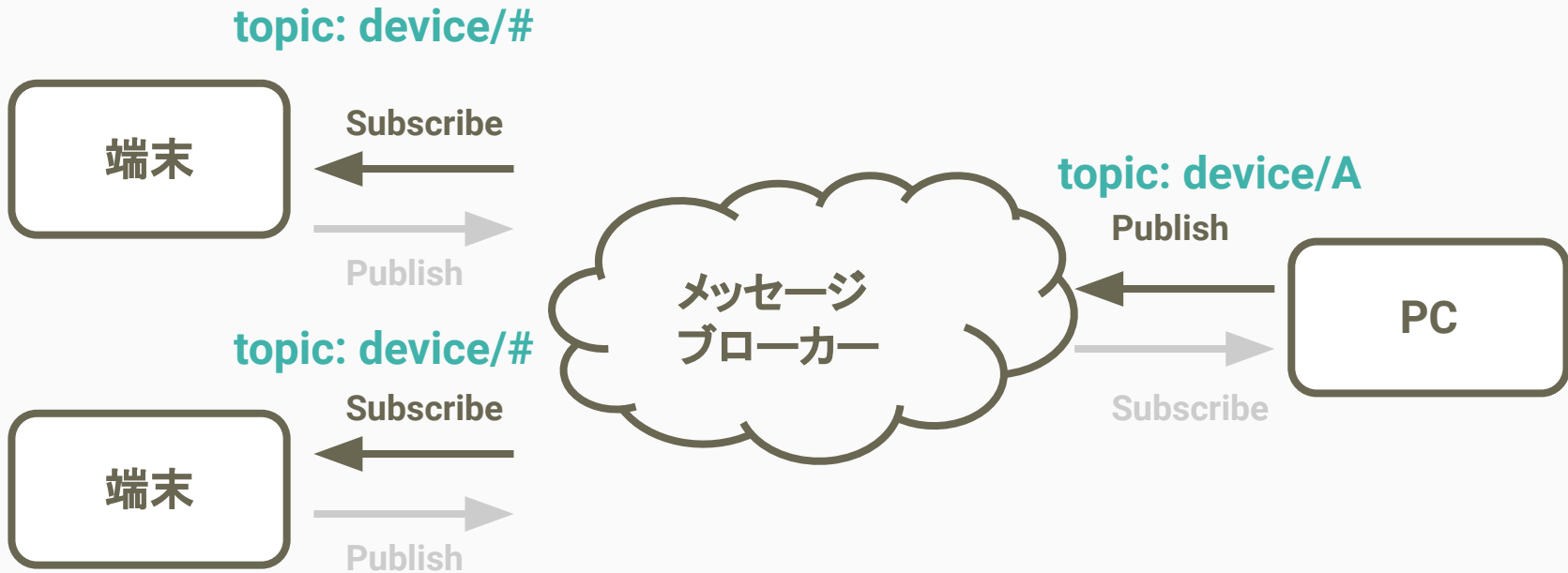
- 端末とAWSとの双方向通信を可能にする
- 端末をThingで管理する
- MQTT通信を使って端末と通信
- グループでThingを管理
- 他のAWSサービスとの連携
- などなど

MQTT

- 軽量なプロトコル、Pub/Subモデル
- Topic単位でPublish/Subscribeする
 - Topic例: device/abc/def
- SubscribeしてるTopicにPublishされたメッセージのみ受信する
- AWS IoTでは基本コレでメッセージをやりとりする







AWS IoTを使った管理

主な機能

- 端末の単純なリモート操作
- 端末状態の取得
- アプリのアップデート

端末の単純なリモート操作

端末の単純なリモート操作

- 管理画面から再起動、初期化などの端末操作
- AWS IoTのTopicメッセージを使用
- 端末は受け取ったメッセージから処理を実行するだけ



Subscribe:
cmd/reboot/thing1



Subscribe:
cmd/reboot/thing2



AWS IoT Core

Publish:
cmd/reboot/thing1



管理画面

端末状態の取得

端末状態の取得

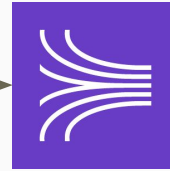
- 1時間に1回、端末から状態をPublish
 - バージョン、バッテリー状態など
- AWS IoTルールを使い、KinesisとLambdaを経由してDynamoDBを保存
- DynamoDBの内容を管理画面で表示



Publish:
data/device-info/A



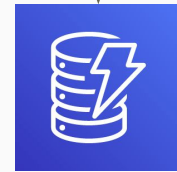
AWS IoT Core



Amazon Kinesis



AWS Lambda



Amazon DynamoDB

ルールアクション実行

1時間に1回 Publish

- ・OS/アプリバージョン
- ・バッテリー状態
- ・などなど

AWS IoT ルール

- MQTTのメッセージをトリガーに他のAWSサービスと連携することができる
- Topicやメッセージの内容を条件にすることが可能
 - 条件はSQLのSELECTのような形式で指定する
 - 例: *SELECT * FROM 'topic/sample/#' WHERE value > 50*

ルール

有効

アクション ▾

概要

説明

編集

Tags

デバイス情報を受け取りKinesisへ流します。KinesisからLambdaを実行し、最終的にDynamoDBへ保存します。

ルールクエリステートメント

編集

このルールを使用して処理するメッセージのソース。

```
SELECT * FROM 'data/device-info/#'
```

SQL バージョンの使用 2016-03-23

アクション

アクションは、ルールがトリガーされたときの処理を表します。 [詳細はこちら](#)



Amazon Kinesis ストリームにメッセージを送信...
store-tv-development-device-info

削除

編集 ▶

アクションの追加

アプリのアップデート

アプリのアップデート

- アップデートファイルは事前にS3に
- AWS IoT Jobを使ってダウンロードURLを配布
- アプリ側は配布されたURLからダウンロードしてアップデート

AWS IoT Job

- Thingやグループに対してリモート操作を行う
- 端末は特別なTopicをSubscribeする
 - ***`$aws/things/<thing-name>/jobs/notify-next`***
- Jobのステータスを更新して状態を確認
 - QUEUED, IN_PROGRESS, SUCCEEDED, FAILED など
- 未完了のJobは再取得可能

ジョブ

完了済み

アクション ▾

概要

最終更新日 8月 04, 2020, 15:38:33 (UTC+0900)

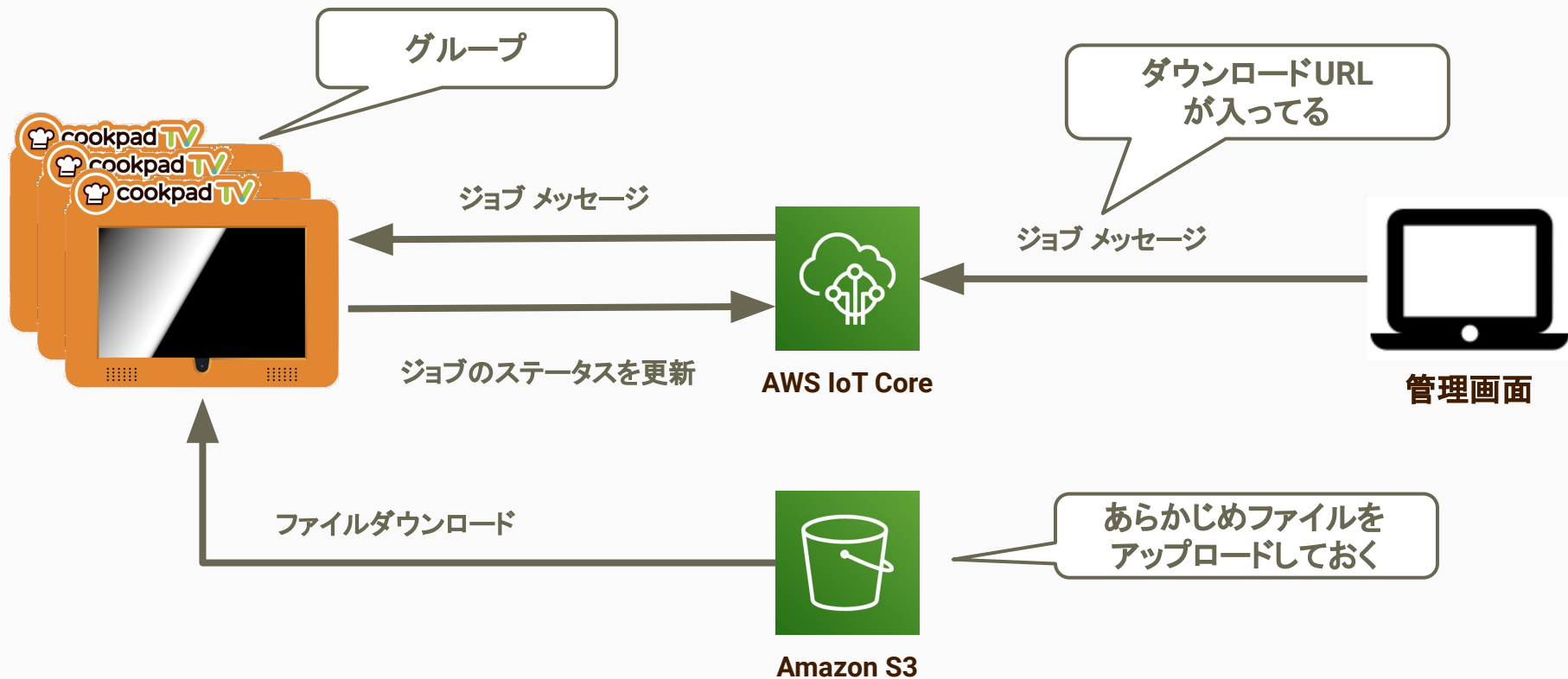
[すべてのステータス](#) [更新](#)

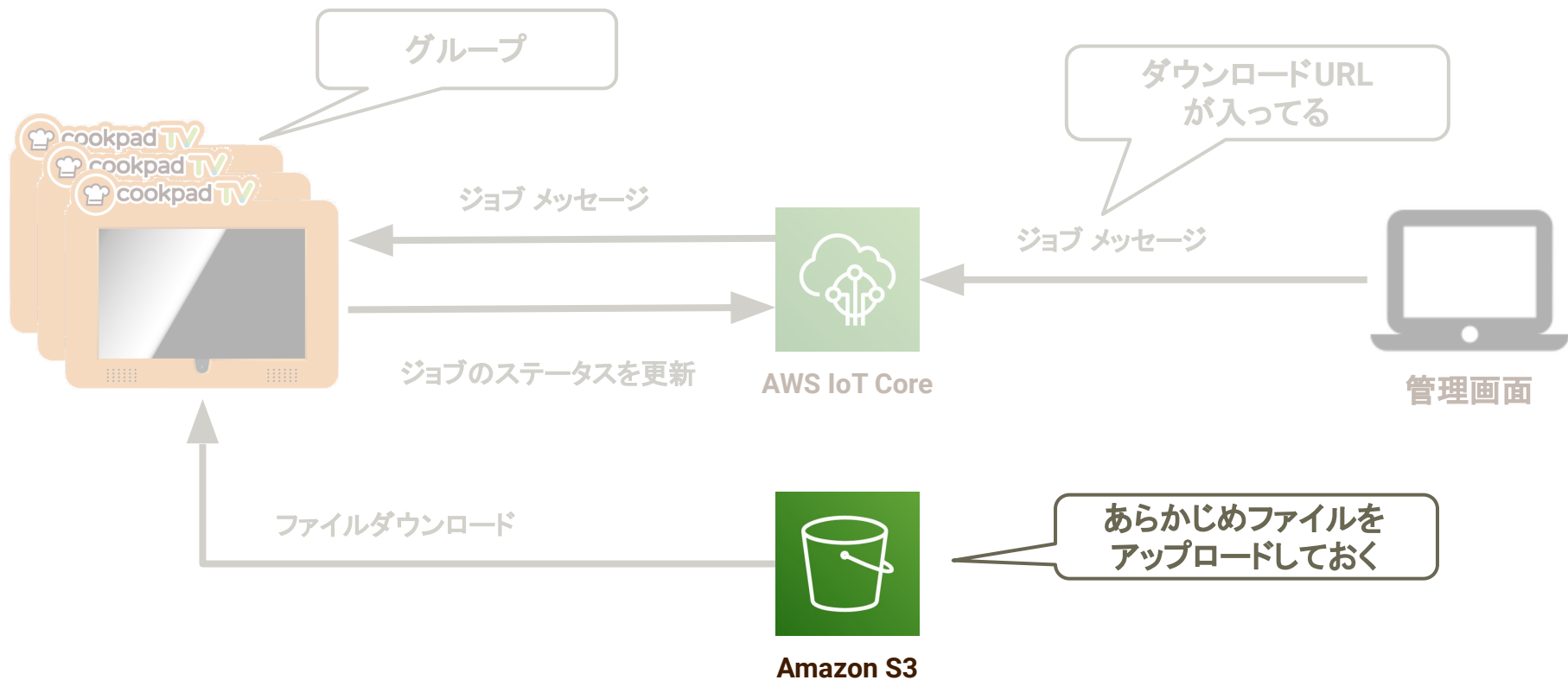
詳細

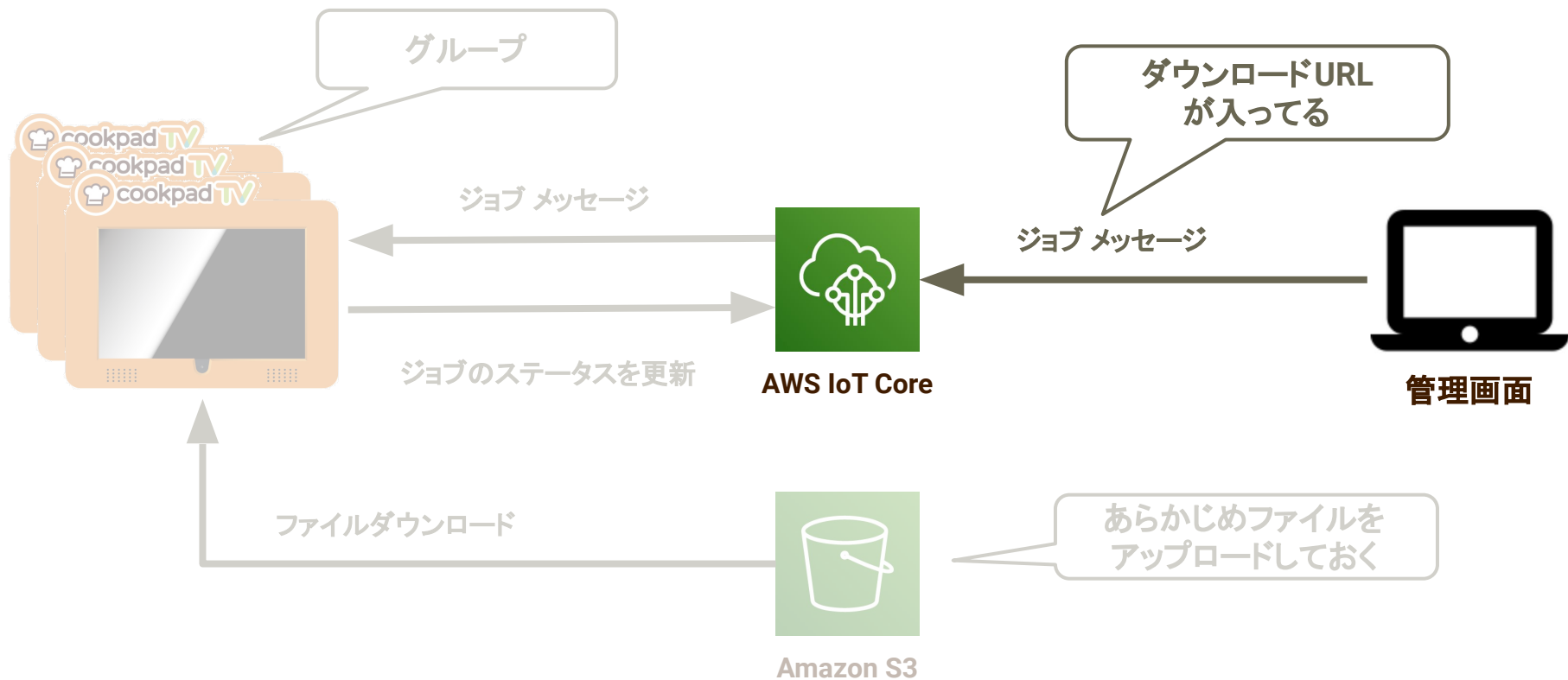
リソースタグ

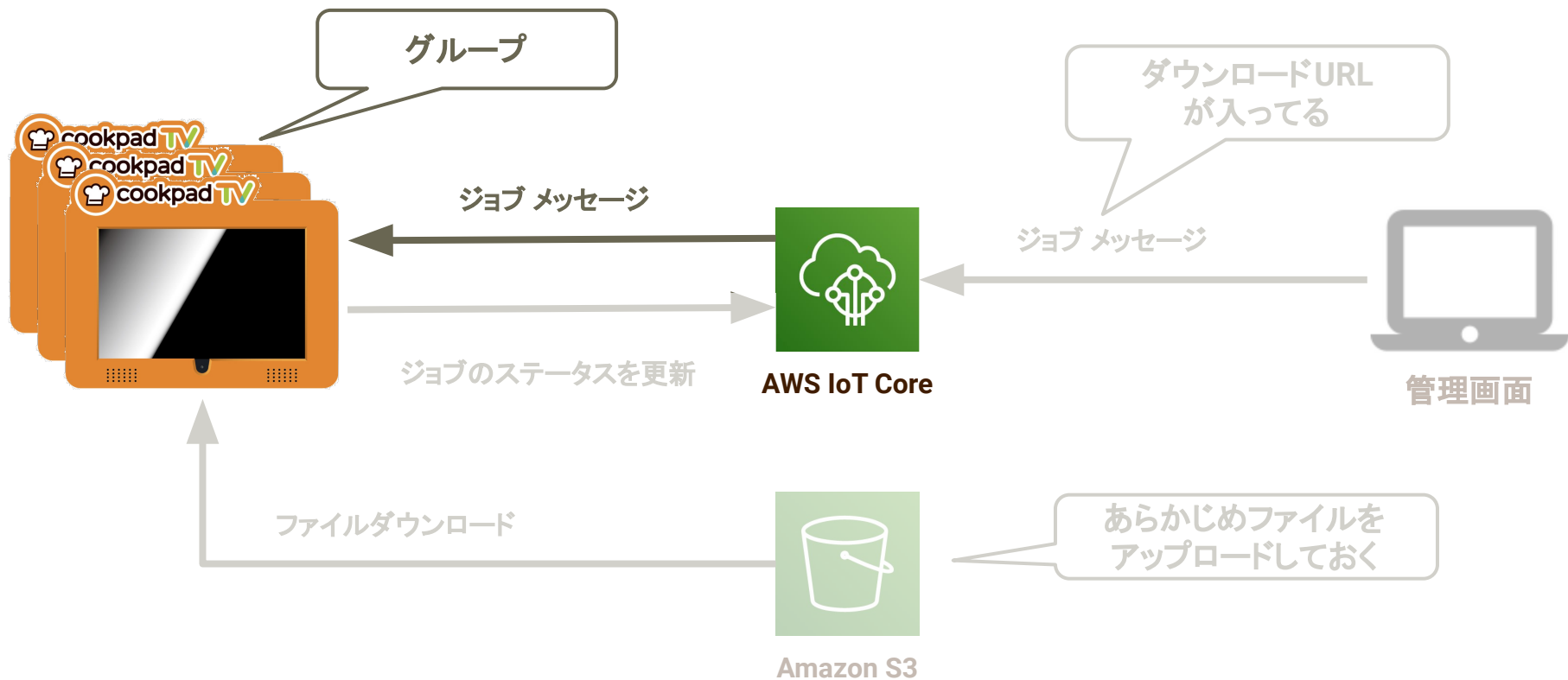
0 キュー	0 進行中	0 タイムアウト	0 失敗	4 成功	0 却下	0 キャンセル済み	0 削除済み
----------	----------	-------------	---------	---------	---------	--------------	-----------

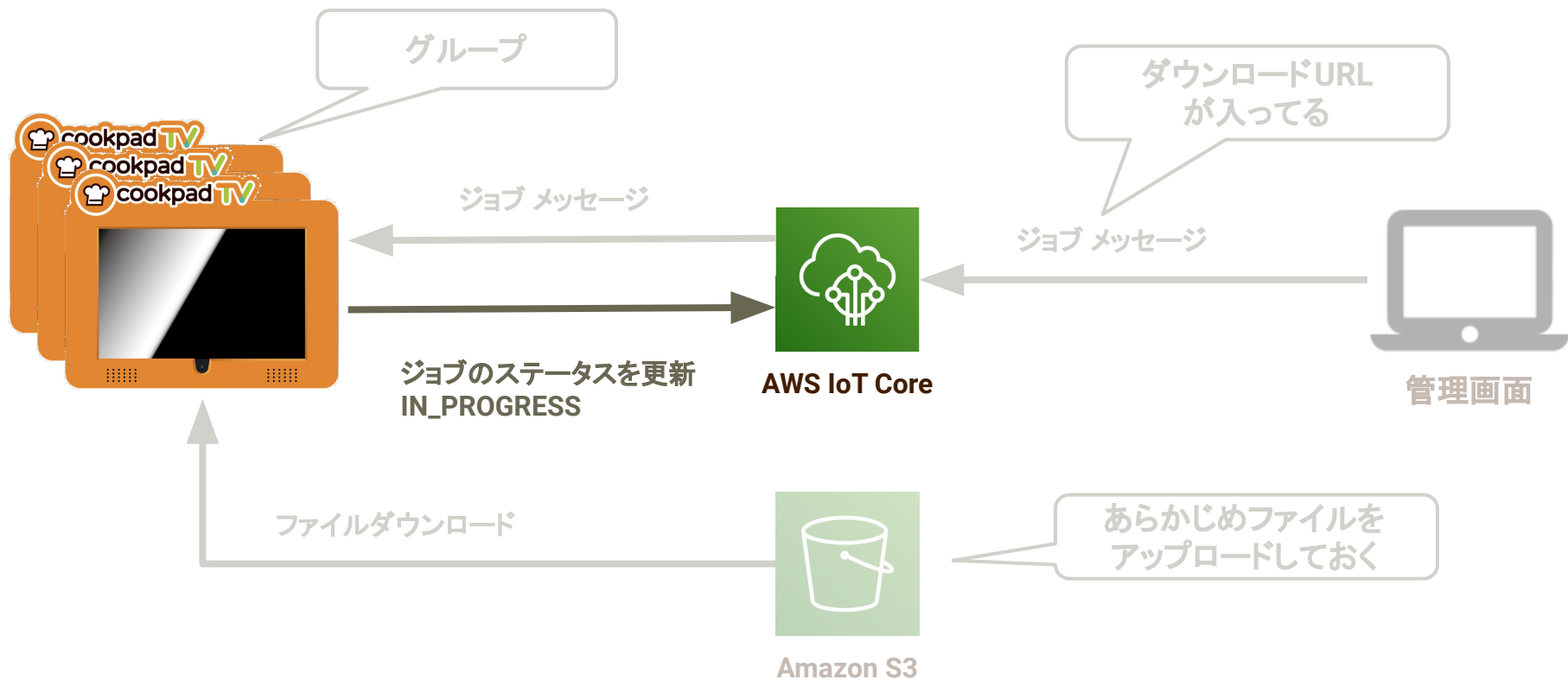
リソース	最終更新日	ステータス
> [redacted]	7月 30, 2020, 14:50:04 (UTC+0...	成功
> [redacted]	7月 30, 2020, 14:50:02 (UTC+0...	成功
> [redacted]	7月 30, 2020, 14:49:57 (UTC+0...	成功
> [redacted]	7月 30, 2020, 14:49:55 (UTC+0...	成功

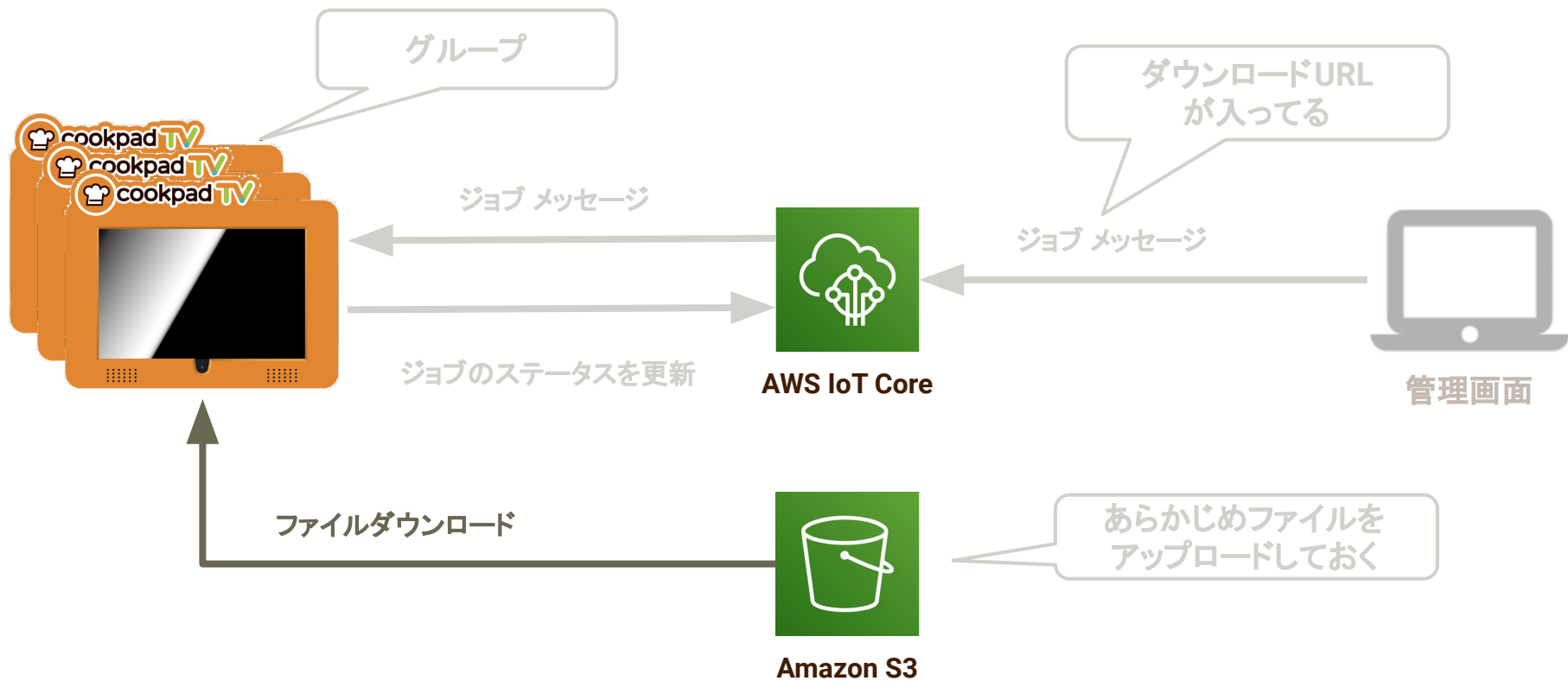


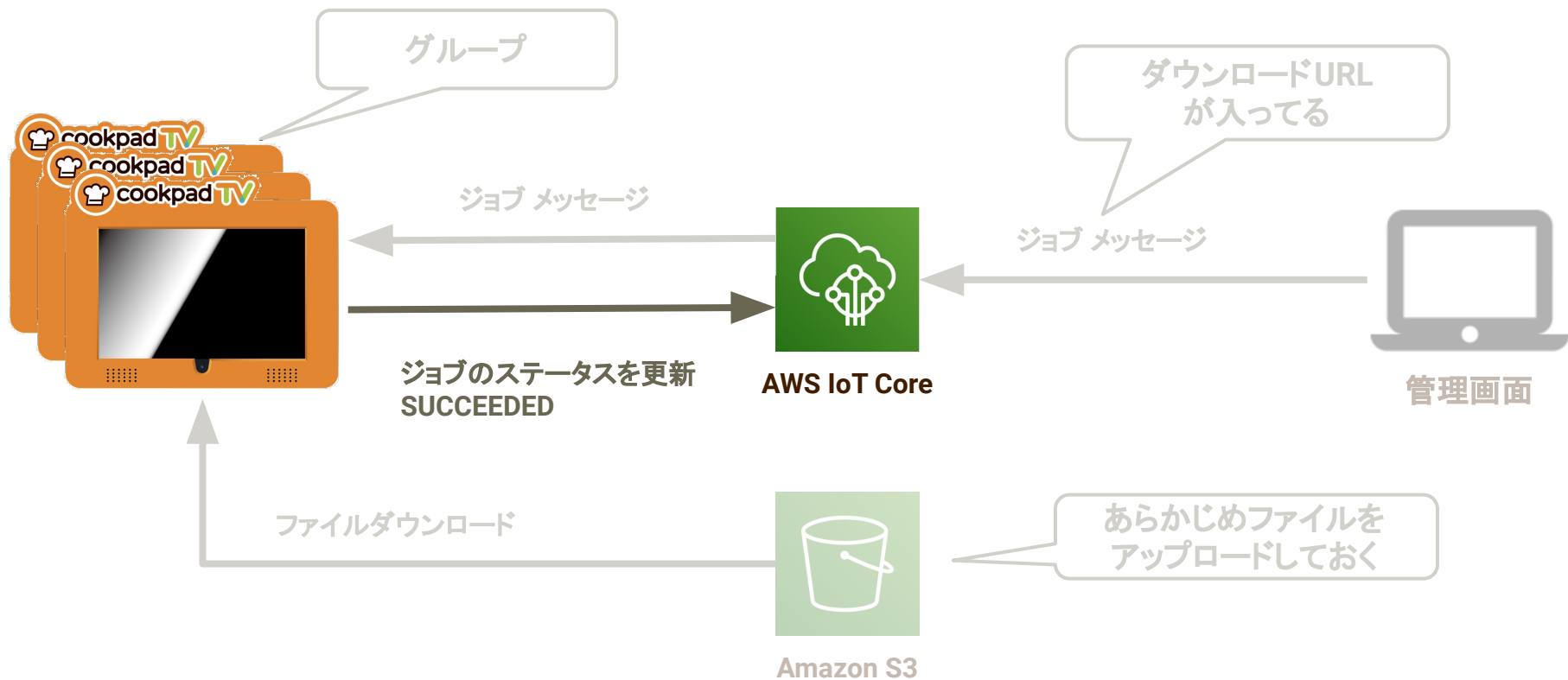












これから

ありがとうございました